UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/633,037	08/04/2000	Yun-Wu Huang	YOR9-1999-0457-US1	2338

7590 03/01/2005

William E Lewis
Ryan & Mason LLP
90 Forest Avenue
Locust Valley, NY 11560

EXAMINER	
STEELMAN, MARY J	
ART UNIT	**PAPER NUMBER**
2122	

DATE MAILED: 03/01/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/633,037	HUANG ET AL.	
	Examiner	Art Unit	
	Mary J. Steelman	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *8/4/00,12/22/2000,3/14/01,7/30/04*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-13* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 September 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to Applicant's request on 28 July 2004 for election of claims 1-13, with traverse, canceling claims 14-47 without prejudice. Claims 1-13 are pending.

Election/Restrictions

2. Applicant is reminded that upon the cancellation of claims to a non-elected invention, the inventorship must be amended in compliance with 37 CFR 1.48(b) if one or more of the currently named inventors is no longer an inventor of at least one claim remaining in the application. Any amendment of inventorship must be accompanied by a request under 37 CFR 1.48(b) and by the fee required under 37 CFR 1.17(i).

Drawings

3. **INFORMATION ON HOW TO EFFECT DRAWING CHANGES**

Replacement Drawing Sheets

Drawing changes must be made by presenting replacement sheets which incorporate the desired changes and which comply with 37 CFR 1.84. An explanation of the changes made must be presented either in the drawing amendments section, or remarks, section of the amendment paper. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). A replacement sheet must include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of the amended drawing(s) must not be labeled as "amended." If the changes to the drawing figure(s) are not

accepted by the examiner, applicant will be notified of any required corrective action in the next Office action. No further drawing submission will be required, unless applicant is notified.

Identifying indicia, if provided, should include the title of the invention, inventor's name, and application number, or docket number (if any) if an application number has not been assigned to the application. If this information is provided, it must be placed on the front of each sheet and centered within the top margin.

Annotated Drawing Sheets

A marked-up copy of any amended drawing figure, including annotations indicating the changes made, may be submitted or required by the examiner. The annotated drawing sheet(s) must be clearly labeled as "Annotated Sheet" and must be presented in the amendment or remarks section that explains the change(s) to the drawings.

Timing of Corrections

Applicant is required to submit acceptable corrected drawings within the time period set in the Office action. See 37 CFR 1.85(a). Failure to take corrective action within the set period will result in ABANDONMENT of the application.

If corrected drawings are required in a Notice of Allowability (PTOL-37), the new drawings MUST be filed within the THREE MONTH shortened statutory period set for reply in the "Notice of Allowability." Extensions of time may NOT be obtained under the provisions of 37 CFR 1.136 for filing the corrected drawings after the mailing of a Notice of Allowability.

New corrected drawings in compliance with 37 CFR 1.121(d) are required in this application because drawings are informal. Applicant is advised to employ the services of a competent patent draftsperson outside the Office, as the U.S. Patent and Trademark Office no longer prepares new drawings. The corrected drawings are required in reply to the Office action to avoid abandonment of the application. The requirement for corrected drawings will not be held in abeyance.

4. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description:
Fig. 9, #901 is not mentioned in the Specification.
Fig. 14, #1402, is referred to as #402 in the Specification. See page 21, line 13. Change the Specification to read '1420'.
Fig. 18, #1805, #1806 are not mentioned in the Specification.
Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet

submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

5. The use of the trademark JAVA / VISUAL BASIC has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Claim Objections

6. Claims 9, 10, 12, and 13 recite "The software system of clam 6...", should be –The software system of claim 6...-- Add an 'i' to clam.

Claims 3, 4, 5, and 6 recite the limitations: "web browser context-based operating system resources", "non-web browser-based operating system resources", "web browser context-based operating system resources', and "non-web browser-based operating system resources". For clarity purposes all references to this feature should be uniform, as recited in claim 1: "non-web browser context-based operating system resources" (meaning resources of the operating system environment) and "web browser context-based operating system resources" (meaning resources limited to the browser environment).

Claim Rejections - 35 USC § 112

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 3, 5, 10, and 13 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 10 recites the limitation "operating system application programming interfaces" in lines 9 and 10, page 28.

Claim 13 recites the limitation "the data language modeling language parser" in line 17, page 28.

There is insufficient antecedent basis for these limitations in the claims.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-7, 10, and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,101,539 to Kennelly et al., in view of US Patent 6,275,938 B1 to Bond et al.

Per claim 1:

A method of generating a web-based application, the method comprising the steps of:

(Col. 1, lines 38-47, "The switch accepts an interface request from a user...and establishes a request processor...to construct a set of management objects.", col. 1, lines 55-60, "The management objects can render a graphical user interface (generate a web-based application)...", col. 2, lines1-6, "The request processor object parses data files and dynamically produces management objects...The request processor object uses indicator tags to parse the data files. The indicator tags can include...a script tag.", col. 20, lines 7-21, "A method..."

-composing one or more web pages in accordance with a scripting language to form the application;

Kennelly disclosed, (col. 6, lines 52), "Web pages are written with the embedded scripts (scripting language) that enable the database management subsystem to provide queries to the operating system and schema database to access objects..." Col. 7, lines 4-6, "The scripting parser will combine the returned objects with HTML to complete a web page to present on web browser."

-embedding one or more extended function calls in the one or more web pages in accordance with the scripting language such that the application,

Kennelly disclosed (col. 6, lines 40-48), "...a script in the web page can provide that a view furnished to the user show the user all...The embedded script (an embedded script with added functionality) is fed to the scripting parser to access a portion of the database..." Kennelly failed

to disclose specifically "when executed by a computer system in which the application is installed, has access to one or more non-web browser context-based operating system resources associated with the computer system through the one or more extended function calls." However, Bond provided more details regarding embedded functionality. Embeded functionality in the form of (col. 4, line 50) Applets, controls, or ActiveX controls. Col. 5, lines 64-67, "FIG. 4 illustrates the major steps...that allow applets (or controls) ...to access all of the operating system services..." Col. 6, lines 35-36, "The thunk DLLs block or restrict many APIs that are not considered safe." It is well known that code that accesses the system resources may be unsafe. Bond selectively allows the code to execute by using (col. 7, lines 19-21) 'sniff code' to examine all memory reads and writes prior to allowing or disallowing access.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to modify Kennelly's invention which composes web pages and embeds extended function calls, to include Bond's invention disclosing embedded controls and a manner to safely execute them because scripted web applications with embedded functionality should have a technique for ensuring that the executed code will be safe, will not corrupt system resources, inject viruses, Trojan horses or otherwise enable malicious code.

Per claim 2:

-the one or more embedded extended function calls cause one or more operating system application programming interfaces to be executed in order to access the one or more non-web browser context-based operating system resources.

Kennelly failed to disclose specifically an embedded extended function call to an API to access a non-web browser context-based operating system resource. However, Bond did disclose a technique for downloading a control (col. 8, lines 33-34), and depending on the security related information available, decides whether to execute it (col. 8, lines 40-45.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to modify Kennelly's invention which composes web pages and embeds extended function calls, to include Bond's invention disclosing embedded controls and a manner to safely execute them because scripted web applications with embedded functionality should have a technique for ensuring that the executed code will be safe, will not corrupt system resources, inject viruses, Trojan horses or otherwise enable malicious code.

Per claim 3:

-the scripting language includes code for accessing one or more web browser context-based operating system resources associated with the computer system.

(Col. 6, lines 34-39, "A scripting parser is a scripting engine that uses a scripting language that converts embedded scripting from HTML pages into HTML for the browser. The final HTML that is sent to the browser may contain data that was retrieved from objects in the object manager." Content within the browser environment (web browser context-based operating system resources) is processed.)

Per claim 4:

Apparatus for processing a web-based application in accordance with a computer system, the apparatus comprising:

-at least one processor operative to:

(See FIG. 1 – network computer system (processor implied in server) FIG. 2 – database management subsystem.

(i) obtain the web-based application, wherein the application is composed of one or more web pages;

Kennelly disclosed, web-based applications, web pages and embedded scripts: FIG. 2, #40 HTML embedded script, Col. 10, lines 46-50, "...the user requests a management object...Through the web page, the user requests the management object...The management object request processor processes the request (management object request processor obtains the web-based application)...", col. 11, lines 15-17, "management object request processor provides management objects to provide a user-specific interface...", col. 11, lines 25-26, "the data manager searches for ...scripts and tags", col. 11, lines 31-35, "If the data manager encounters information in the data file that must be parsed, e.g., scripts, the data manager transfers the information to the management object request processor", col. 12, lines 22-224, "data manager scans data files for scripts...transfers the scripts to the parser..."Col. 12, line 33, "the parser reads the script...", col. 12, lines 36-38, "The function (access to one or more non-web browser context-based operating system resources) may be called as an object, or the function may be embedded in the script that the parser is examining."

Kennelly failed to disclose specifically that the script "has access to one or more non-web browser-based operating system resources associated with the computer system". However Bond disclosed Applets, controls, or ActiveX controls that may access non-web browser-based operating system resources, with consideration given to the security of the access. See response in claim 1.

(ii) interpret scripting language associated with the one or more web pages of the application; (Col. 5, lines 10-13, "The object manager parses the string passed to it,...to further interpret the string.", col. 12, lines 26-28, "The parser parses the scripts and subsequently organizes a response based on the scripts (interprets the scripting language).

(iii) convert one or more calls embedded in the interpreted scripting language associated with the one or more web pages into code that executes one or more application programming interfaces so as to access the one or more non-web browser-based operating system resources associated with the computer system.

Kennelly failed to disclose accesses to non-web browser-based operating system resources. However, Bond disclosed that 'untrusted code', code that can access the system resources, may be selectively executed (convert embedded code into executable) depending on the security information available.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to modify Kennelly's invention which composes web pages and embeds extended function calls, to include Bond's invention disclosing embedded controls and a manner to safely execute them because scripted web applications with embedded functionality should have a technique for ensuring that the executed code will be safe, will not corrupt system resources, inject viruses, Trojan horses or otherwise enable malicious code.

Per claim 5:

-the at least one processor is further operative to interpret scripting language code for accessing one or more web browser context-based operating system resources (non-web browser context-based operating system resources) associated with the computer system.

(Col. 5, lines 10-13, "The object manager parses the string passed to it,...to further interpret the string.", col. 12, lines 26-28, "The parser parses the scripts and subsequently organizes a response based on the scripts (interprets the scripting language within the browser environment).

Per claim 6:

A software system for processing a web-based application in accordance with a computer system, the software system comprising:

(Col. 2, lines 48-54, "FIG. 1, a network system architecture...runs a web browser..."

-an application manager that manages a life-cycle associated with the web-based application in accordance with the computer system, wherein the application is composed of one or more web

pages and has access to one or more non-web browser-based operating system resources associated with the computer system;

(See FIGs. 1 & 2. Col. 1, lines 42-43, "accepts an interface request (web-based application / web pages) from a user...", col. 11, line 66 – col. 12, line 12, "If the data manager encounters one of the script tags..."noc_tag func", the data manager transfers the script (convert one or more calls embedded in the interpreted scripting language associated with the one or more web pages) that lies between the tag and the end tag...to the parser...If the data manager encounters the server side include tag...noc_inc, the data manager calls the associated file name... (depending on the type of tag encountered, the script may call for accessing a function or a system file / accesses operating system resources)", col. 3, lines 13-17, "A web browser interfaces with the database management software by using a graphical user interface (GUI). One preferred GUI is provided by using hypertext markup language (HTML) based web pages."

Kennelly failed to specifically disclose access to one or more non-web browser-based operating system resources. However, Bond disclosed that an embedded Applet, control or ActiveX control may be given access to system resources, depending on security information provided. Col. 6, lines 53-57, "Some APIs may be allowed under some conditions..."

-a scripting language interpreter that interprets scripting language associated with the one or more web pages of the application;

(Col. 5, lines 10-13, "The object manager parses the string passed to it...further interpret the string..."

-an operating system interface module which converts one or more calls embedded in the interpreted scripting language associated with the one or more web pages into code that executes one or more application programming interfaces...

Kennelly disclosed, FIG. 7, #152, 'Management Object', col. 11, lines 15-17, "The management object request processor provides management objects (operating system interface module) to provide a user-specific interface...", col. 11, lines 22-23, "the data manager and the parser analyze the data files..." col. 11, line 66 – col. 12, line 12, "If the data manager encounters one of the script tags..."noc_tag func" (embedded call), the data manager transfers the script that lies between the tag and the end tag...to the parser...If the data manager encounters the server side include tag...noc_inc, the data manager calls the associated file name... (depending on the type of tag encountered, the script may call for accessing a function or a system file / accesses operating system resources)", col. 3, lines 13-17, "A web browser interfaces with the database management software by using a graphical user interface (GUI). One preferred GUI is provided by using hypertext markup language (HTML) based web pages."

Kennelly failed to disclose, "to access the one or more non-web browser-based operating system resources associated with the compute system." However, Bond disclosed that a Applet, control, or ActiveX control downloaded (col. 4, lines 30-34) from a web page (link embedded in a script)

could be permitted to access system resources, depending on security issued involved (col. 5, lines 65-67).

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to modify Kennelly's invention which composes web pages and embeds extended function calls, to include Bond's invention disclosing embedded controls and a manner to safely execute them because scripted web applications with embedded functionality should have a technique for ensuring that the executed code will be safe, will not corrupt system resources, inject viruses, Trojan horses or otherwise enable malicious code.

Per claim 7:

-a web browser to at least one of retrieved web objects, send web requests, and provide a graphical user interface in accordance with the execution of the web-based application.

(Col. 3, lines 13-17, "A web browser interfaces with the database management software by using a graphical user interface (GUI). One preferred GUI is provided by using hypertext markup language (HTML) based web pages", col. 12, lines 33-55, "The parser reads the script...calls for the function. The function may be called as an object or the function may be embedded in the script...parser may request data from the object manager (web requests)...or set data in the object manager...parser typically incorporates the data from the object manager into a portion of the GUI...response from the parser are output as a Web page on the GUI (provide a graphical user interface in accordance with the execution of the web-based application)."

Per claim 10:

-the scripting language interpreter parses the one or more calls embedded in the scripting language and the application manager executes the one or more parsed calls in association with the operating system interface module…

Kennelly disclosed, FIG. 7, #152, 'Management Object', col. 11, lines 15-17, "The management object request processor provides management objects to provide a user-specific interface…", col. 11, lines 22-23, "the data manager and the parser (scripting language interpreter) analyze (parse) the data files…" col. 11, line 66 – col. 12, line 12, "If the data manager encounters one of the script tags…"noc_tag func" (embedded call), the data manager transfers the script that lies between the tag and the end tag…to the parser…If the data manager encounters the server side include tag…noc_inc, the data manager calls the associated file name… (depending on the type of tag encountered, the script may call for accessing a function or a system file / invoke code associated with operating system resources)", col. 3, lines 13-17, "A web browser interfaces with the database management software by using a graphical user interface (GUI). One preferred GUI is provided by using hypertext markup language (HTML) based web pages.", col. 12, lines 33-55, "The parser reads the script (scripting language interpreter parses) …calls for the function. The function may be called as an object or the **function may be embedded in the script**…parser may request data from the object manager …or set data in the object manager…parser typically incorporates the data from the object manager into a portion of the GUI…response from the parser are output as a Web page on the GUI…"

Kennelly failed to disclose, "to invoke code associated with the operating system application programming interfaces for the one or more calls." However, Bonds disclosed embedded links to Applets, controls and ActiveX controls, that when downloaded (col. 4, lines 30-34 & 50-51) are selectively executed depending on their related security issued (col. 5, lines 51-63).

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to modify Kennelly's invention which composes web pages and embeds extended function calls, to include Bond's invention disclosing embedded controls and a manner to safely execute them because scripted web applications with embedded functionality should have a technique for ensuring that the executed code will be safe, will not corrupt system resources, inject viruses, Trojan horses or otherwise enable malicious code.

Per claim 13:

-a web browser and a data modeling language parser, wherein the application manager is operative to:

(i) process code in each web page of the application

(Col. 3, lines 13-17, col. 4, lines 10-16.)

(ii) invoke the web browser to process code that is of a visual presentation type;

(Col. 7, lines 4-7, "The scripting parser will combine the retuned objects with HTML to complete a web page to present on web browser." It is inherent that a web browser visually presents the code.)

(iii) invoke the data modeling language parser to parse code that is of a data modeling language type;

Kennelly disclosed (col. 7, lines 4-7), "The scripting parser will combine the retuned objects with HTML to complete a web page to present on web browser."

XML is a type of markup language used to facilitate the interchange of data over a network. Both XML and HTML are derived from SGML, a mark up language standard provided by the World Wide Web Consortium (W3C). XML provides information related to the organization of stored of data, such as in a database or table. It is inherent that XML is used when a user request to access a database or table for information (col. 17, lines 60-62) is processed, as may occur depending on user privileges.

(iv) invoke the scripting language interpreter to parse code that is of the scripting language type such that the scripting language interpreter may execute code that is of the original scripting language type used by the web browser;

(Col. 7, lines 4-7, "The scripting parser will combine the retuned objects with HTML to complete a web page to present on web browser." It is inherent that a web browser visually presents the code.)

(v) invoke the operating system interface module to execute code, successfully parsed by the scripting language interpreter, that executes the one or more application programming interfaces.

Kennelly failed to specifically disclose instances of embedded functionality accessing an operating system interface. However, Bonds disclosed, (col. 4, lines 30-32) downloading embedded code, in the form of an Applet, control, or ActiveX control, that may use an API call to access system resources (col. 5, line 51). It is known that such types of actions can be (col. 5, line 55) 'a program of uncertain provenance or effects, which might damage the system..." Bond provides a technique for selectively permitting the access depending on the security issues. Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to modify Kennelly's invention which composes web pages and embeds extended function calls, to include Bond's invention disclosing embedded controls and a manner to safely execute them because scripted web applications with embedded functionality should have a technique for ensuring that the executed code will be safe, will not corrupt system resources, inject viruses, Trojan horses or otherwise enable malicious code.

11. Claims 8, 9, 11, and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,101,539 to Kennelly et al., in view of US Patent 6,275,938 B1 to Bond et al., and further in view of US Pre Grant Pub 20030154279 A1 to Aziz .

Per claim 8:

Kennelly disclosed (col. 7, lines 4-6), "The scripting parser will combine the returned objects with HTML to complete a web page to present on web browser." Kennelly failed to disclose specifically, "a data modeling language parser to decode information in the web-based application written in a corresponding data modeling language."

However, Aziz disclosed [0135], " Servlet 412 passes FEML text 410' from server 404 to client 402, which stores FEML text 410. A JavaScript XML Parser process 424 is applied to FEML text 410, resulting in creating and storing FEML object model 408. Client 402 carries out a Generate-Farm process on FEML object model 408, resulting in creating and storing JavaScript object model 406, which may be imported into and manipulated by the client editor." It is well known that XML is an extended markup language, commonly used to define the schema and allow for data base transactions.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify the Kennelly / Bond combination by more specifically disclosing that XML may be used as a data modeling language, and processed with an XML parser because XML is an extended markup language, commonly used to define the schema and allow for data base transactions when format is not required to be in a uniform pattern.

Per claim 9:

-the data modeling language parser is an XML parser.

See rejection of claim 8 above. It is inherent that an extended mark up language, such as XML, is defined in a document and provided with tags relative to the data involved, and a parser is able to interpret the data by using the definition document.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify the Kennelly / Bond combination by more specifically disclosing that XML may be used as a data modeling language, and processed with an XML parser because XML is an extended markup language, commonly used to define the schema and allow for data base transactions when format is not required to be in a uniform pattern.

Per claim 11:

-the scripting language is JavaScript.

See rejection of claim 8 above. It is inherent that the scripting language could be JavaScript, a Sun trademarked version of script.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify the Kennelly / Bond combination by more specifically disclosing that XML may be used as a data modeling language, and processed with an XML parser because XML is an extended markup language, commonly used to define the schema and allow for data base transactions when format is not required to be in a uniform pattern.

Per claim 12:

-the scripting language interpreter is a JavaScript interpreter.

Kennelly disclosed (col. 7, lines 4-7), "The scripting parser will combine the retuned objects with HTML to complete a web page to present on web browser", but failed to explicitly state that a JavaScript interpreter was used.

See rejection of claim 8 above. It is inherent that a JavaScript specific interpreter be used to interpret JavaScript, just as another suitable interpreter would be used to interpret another version of script.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify the Kennelly / Bond combination by more specifically disclosing that XML may be used as a data modeling language, and processed with an XML parser because XML is an extended markup language, commonly used to define the schema and allow for data base transactions when format is not required to be in a uniform pattern.

Conclusion

12. XML is a type of markup language used to facilitate the interchange of data over a network. Both XML and HTML are derived from SGML, a mark up language standard provided by the World Wide Web Consortium (W3C). XML provides information related to the organization of stored of data, such as in a database or table. It should be noted that HTML, XML, scripting, JavaScript (a scripting language trademarked by Sun Microsystems, Inc.), and embedded functionality in the form of controls or plugins are well known in the art. Generally, embedded controls or functionality that access system resources are not desirable due to their ability to corrupt the system. As such, the JAVA programming language provides a 'sandbox' to isolate the code from the system, thereby providing a degree of security.

13. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached at (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

02/10/2005

TUAN DAM
SUPERVISORY PATENT EXAMINER